P⊙RTAL

USPTO

Search: ⊙ The ACM Digital Library   ○ The Guide

conflict misses   **SEARCH**

THE ACM DIGITAL LIBRARY

ī Feedback Report a problem Satisfaction survey

Terms used **conflict misses**   Found **22,268** of **161,645**

Sort results by	relevance ▾	📎 Save results to a Binder	Try an Advanced Search
Display results	expanded form ▾	[?] Search Tips	Try this search in The ACM Guide
		☐ Open results in a new window	

Results 1 - 20 of 200   Result page: **1** 2 3 4 5 6 7 8 9 10   next
Best 200 shown   Relevance scale ☐▭◼◼◼

1 Runtime identification of cache conflict misses: The adaptive miss buffer ◼
Jamison D. Collins, Dean M. Tullsen
November 2001 **ACM Transactions on Computer Systems (TOCS)**, Volume 19 Issue 4

Full text available: 📄 pdf(1.08 MB)   Additional Information: full citation, abstract, references, index terms

This paper describes the miss classification table, a simple mechanism that enables the processor or memory controller to identify each cache miss as either a conflict miss or a capacity (non-conflict) miss. The miss classification table works by storing part of the tag of the most recently evicted line of a cache set. If the next miss to that cache set has a matching tag, it is identified as a conflict miss. This technique correctly identifies 88% of misses.Several applications of this i ...

Keywords: Cache architecture, adaptive miss buffer, cache exclusion, conflict misses, prefetching, victim cache

2 Hardware identification of cache conflict misses ◼
Jamison D. Collins, Dean M. Tullsen
November 1999 **Proceedings of the 32nd annual ACM/IEEE international symposium on Microarchitecture**

Full text available: 📄 pdf(1.12 MB) 🗐 Additional Information: full citation, abstract, references, citings, index terms
Publisher Site

This paper describes the Miss Classification Table, a simple mechanism that enables the processor or memory controller to identify each cache miss as either a conflict miss or a capacity (non-conflict) miss. The miss classification table works by storing part of the tag of the most recently evicted line of a cache set. If the next miss to that cache set has a matching tag, it is identified as a conflict miss. This technique correctly identifies 87% of misses in the worst case.

3 Eliminating conflict misses for high performance architectures ◼
Gabriel Rivera, Chau-Wen Tseng
July 1998 **Proceedings of the 12th international conference on Supercomputing**

Full text available: 📄 pdf(1.04 MB)   Additional Information: full citation, references, citings, index terms

4   ◼

Data transformations for eliminating conflict misses

Gabriel Rivera, Chau-Wen Tseng
May 1998 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1998 conference on Programming language design and implementation**, Volume 33 Issue 5

Full text available: 📄 pdf(1.62 MB)   Additional Information: full citation, abstract, references, citings, index terms

Many cache misses in scientific programs are due to conflicts caused by limited set associativity. We examine two compile-time data-layout transformations for eliminating conflict misses, concentrating on misses occuring on every loop iteration. Inter-variable padding adjusts variable base addresses, while intra-variable padding modifies array dimension sizes. Two levels of precision are evaluated. PADLITE only uses array and column dimension sizes, relying on assumptions about common array refe ...

5 Avoiding conflict misses dynamically in large direct-mapped caches   ■
Brian N. Bershad, Dennis Lee, Theodore H. Romer, J. Bradley Chen
November 1994 **Proceedings of the sixth international conference on Architectural support for programming languages and operating systems**, Volume 29 , 28 Issue 11 , 5

Full text available: 📄 pdf(1.37 MB)   Additional Information: full citation, abstract, references, citings, index terms

This paper describes a method for improving the performance of a large direct-mapped cache by reducing the number of conflict misses. Our solution consists of two components: an inexpensive hardware device called a Cache Miss Lookaside (CML) buffer that detects conflicts by recording and summarizing a history of cache misses, and a software policy within the operating system's virtual memory system that removes conflicts by dynamically remapping pages whenever large numbers of conflict miss ...

6 Eliminating cache conflict misses through XOR-based placement functions   ■
Antonio González, Mateo Valero, Nigel Topham, Joan M. Parcerisa
July 1997 **Proceedings of the 11th international conference on Supercomputing**

Full text available: 📄 pdf(1.21 MB)   Additional Information: full citation, references, citings, index terms

Keywords: XOR-based placement functions, cache memory, conflict misses

7 Cache: Reducing traffic generated by conflict misses in caches   ■
Pepijn J. de Langen, Ben Juurlink
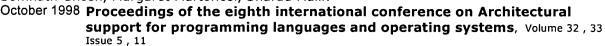April 2004 **Proceedings of the 1st conference on Computing frontiers**

Full text available: 📄 pdf(246.59 KB)   Additional Information: full citation, abstract, references, citings, index terms

Off-chip memory accesses are a major source of power consumption in embedded processors. In order to reduce the amount of traffic between the processor and the off-chip memory as well as to hide the memory latency, nearly all embedded processors have a cache on the same die as the processor core. Because small caches dissipate less power and are cheaper than large caches, a small cache is preferable to a large cache. Furthermore, because set-associative caches consume more power than direct-mapp ...

Keywords: caches, conflict misses, embedded processors, power reduction

8 A permutation-based page interleaving scheme to reduce row-buffer conflicts and exploit data locality   ■
Zhao Zhang, Zhichun Zhu, Xiaodong Zhang

December 2000 **Proceedings of the 33rd annual ACM/IEEE international symposium on Microarchitecture**

Full text available: ⬛ pdf(153.06 KB)

📄 ps(856.21 KB)   Additional Information: full citation, references, citings, index terms

⬛ Publisher Site

---

9 Reducing cache misses using hardware and software page placement   ⬛

Timothy Sherwood, Brad Calder, Joel Emer

May 1999 **Proceedings of the 13th international conference on Supercomputing**

Full text available: ⬛ pdf(1.50 MB)   Additional Information: full citation, references, citings, index terms

---

10 The design and performance of a conflict-avoiding cache   ⬛

Nigel Topham, Antonio González, José González

December 1997 **Proceedings of the 30th annual ACM/IEEE international symposium on Microarchitecture**

Full text available: ⬛ pdf(1.21 MB) ⬛   Additional Information: full citation, abstract, references, citings, index terms
Publisher Site

High performance architectures depend heavily on efficient multi-level memory hierarchies to minimize the cost of accessing data. This dependence will increase with the expected increases in relative distance to main memory. There have been a number of published proposals for cache conflict-avoidance schemes. We investigate the design and performance of conflict-avoiding cache architectures based on polynomial modulus functions, which earlier research has shown to be highly effective at reducing ...

Keywords: cache architecture design, cache storage, conflict miss ratios, conflict-avoiding cache performance, data access cost minimization, high performance architectures, main memory, multi-level memory hierarchies, polynomial modulus functions

---

11 Precise miss analysis for program transformations with caches of arbitrary associativity ⬛

Somnath Ghosh, Margaret Martonosi, Sharad Malik

October 1998 **Proceedings of the eighth international conference on Architectural support for programming languages and operating systems**, Volume 32 , 33
Issue 5 , 11

Full text available: ⬛ pdf(1.67 MB)   Additional Information: full citation, abstract, references, citings, index terms

Analyzing and optimizing program memory performance is a pressing problem in high-performance computer architectures. Currently, software solutions addressing the processor-memory performance gap include compiler-or programmer-applied optimizations like data structure padding, matrix blocking, and other program transformations. Compiler optimization can be effective, but the lack of *precise* analysis and optimization frameworks makes it impossible to confidently make optimal, rather than h ...

---

12 Trading conflict and capacity aliasing in conditional branch predictors   ⬛

Pierre Michaud, André Seznec, Richard Uhlig

May 1997 **ACM SIGARCH Computer Architecture News , Proceedings of the 24th annual international symposium on Computer architecture**, Volume 25 Issue 2

Full text available: ⬛ pdf(1.60 MB)   Additional Information: full citation, abstract, references, citings, index terms

As modern microprocessors employ deeper pipelines and issue multiple instructions per

cycle, they are becoming increasingly dependent on accurate branch prediction. Because hardware resources for branch-predictor tables are invariably limited, it is not possible to hold all relevant branch history for all active branches at the same time, especially for large workloads consisting of multiple processes and operating-system code. The problem that results, commonly referred to as aliasing in the br ...

Keywords: 3 C's classification, aliasing, branch prediction, skewed branch predictor

13 Comparing data forwarding and prefetching for communication-induced misses in shared-memory MPs
David Koufaty, Josep Torrellas
July 1998   **Proceedings of the 12th international conference on Supercomputing**
Full text available: pdf(1.12 MB)   Additional Information: full citation, references, citings, index terms

14 Code placement techniques for cache miss rate reduction
Hiroyuki Tomiyama, Hiroto Yasuura
October 1997 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 2 Issue 4
Full text available: pdf(288.29 KB)   Additional Information: full citation, abstract, references, citings, index terms

In the design of embedded systems with cache memories, it is important to minimize the cache miss rates to reduce power consumption of the systems as well as improve the performance. In this article, we propose two code placement methods (a simplified method and a refined one) to reduce miss rates of instruction caches. We first define a simplified code placement problem without an attempt to minimize the code size. The problem is formulated as an integer linear programming (ILP) problem, ...

Keywords: code placement, instruction cache, integer linear programming

15 Instruction prefetching of systems codes with layout optimized for reduced cache misses
Chun Xia, Josep Torrellas
May 1996 **ACM SIGARCH Computer Architecture News , Proceedings of the 23rd annual international symposium on Computer architecture**, Volume 24 Issue 2
Full text available: pdf(1.65 MB)   Additional Information: full citation, abstract, references, citings, index terms

High-performing on-chip instruction caches are crucial to keep fast processors busy. Unfortunately, while on-chip caches are usually successful at intercepting instruction fetches in loop-intensive engineering codes, they are less able to do so in large systems codes. To improve the performance of the latter codes, the compiler can be used to lay out the code in memory for reduced cache conflicts. Interestingly, such an operation leaves the code in a state that can be exploited by a new type of ...
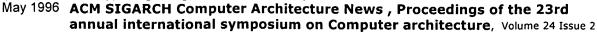
16 Reducing cache conflicts in data cache prefetching
Jin-Ho Lee, Min-Young Lee, Seong-Uk Choi, Myong-Soon Park
September 1994 **ACM SIGARCH Computer Architecture News**, Volume 22 Issue 4
Full text available: pdf(418.71 KB)   Additional Information: full citation, citings, index terms

17 <u>Missing the memory wall: the case for processor/memory integration</u>

Ashley Saulsbury, Fong Pong, Andreas Nowatzyk

May 1996 **ACM SIGARCH Computer Architecture News , Proceedings of the 23rd annual international symposium on Computer architecture**, Volume 24 Issue 2

Full text available: <u>pdf(1.45 MB)</u>   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>

Current high performance computer systems use complex, large superscalar CPUs that interface to the main memory through a hierarchy of caches and interconnect systems. These CPU-centric designs invest a lot of power and chip area to bridge the widening gap between CPU and main memory speeds. Yet, many large applications do not operate well on these systems and are limited by the memory subsystem performance.This paper argues for an integrated system approach that uses less-powerful CPUs that are ...

18 <u>Column-associative caches: a technique for reducing the miss rate of direct-mapped caches</u>

Anant Agarwal, Stephen D. Pudar

May 1993 **ACM SIGARCH Computer Architecture News , Proceedings of the 20th annual international symposium on Computer architecture**, Volume 21 Issue 2

Full text available: <u>pdf(1.17 MB)</u>   Additional Information: <u>full citation</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>

19 <u>Cache miss equations: an analytical representation of cache misses</u>

Somnath Ghosh, Margaret Martonosi, Sharad Malik

July 1997 **Proceedings of the 11th international conference on Supercomputing**

Full text available: <u>pdf(1.98 MB)</u>   Additional Information: <u>full citation</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>

20 <u>Memory optimization for embedded systems: Improved indexing for cache miss reduction in embedded systems</u>

Tony Givargis

June 2003 **Proceedings of the 40th conference on Design automation**

Full text available: <u>pdf(215.59 KB)</u>   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>index terms</u>

The increasing use of microprocessor cores in embedded systems as well as mobile and portable devices creates an opportunity for customizing the cache subsystem for improved performance. In traditional cache design, the index portion of the memory address bus consists of the K least significant bits, where K=log2(D) and D is the depth of the cache. However, in devices where the application set is known and characterized (e.g., systems that execute a fixed application set) there is an opportunity ...

Keywords: cache optimization, design space exploration, index hashing